A robust and adaptive recovery-based discontinuous Galerkin method for the numerical solution of convection-diffusion equations

(Article begins on next page)

26 June 2024

A robust and adaptive recovery-based discontinuous Galerkin method for the numerical solution of convection-diffusion equations

A. Ferrero[1*], F. Larocca[1] and G. Puppo[2]

[1]*Dipartimento di Ingegneria Meccanica e Aerospaziale, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, Italy*
[2]*Dipartimento di Scienza ed Alta Tecnologia, Universitá dell'Insubria, Via Valleggio 11, Como, Italy*

SUMMARY

In this paper we introduce and test the Enhanced Stability Recovery (ESR) scheme. It is a robust and compact approach to the computation of diffusive fluxes in the framework of Discontinuous Galerkin methods. The scheme is characterized by a new recovery basis and a new procedure for the weak imposition of Dirichlet boundary conditions. These features make the method flexible and robust, even in the presence of highly distorted meshes. The implementation is simplified with respect to the original recovery scheme [1] (RDG1x). Furthermore, thanks to the proposed approach, a robust implementation of p-adaptive algorithms is possible. Numerical tests on unstructured grids show a convergence rate equal to $p + 1$, where p is the reconstruction order. Comparisons are shown with the original recovery scheme RDG1x and the widely used method proposed by Bassi, Rebay et al. [2] (BR2). Results show a significantly larger stability region for the proposed discretization when explicit Runge-Kutta time integration is employed. Interestingly, this advantage grows quickly when the reconstruction order is increased. The proposed procedure for the weak imposition of Dirichlet BCs does not need the introduction of ghost cells and it is truly local since it does not require data exchange with other elements. Further, it can be easily used with curvilinear wall elements. Several test cases are considered. They include some benchmark tests with the heat equation and compressible Navier-Stokes equations, and also more demanding problems assessing the behavior of the scheme with very stretched elements and separated flows.
Copyright © 2010 John Wiley & Sons, Ltd.

1. INTRODUCTION

The main idea of the discontinuous Galerkin (DG) methods is to have several degrees of freedom inside each element in order to obtain high order reconstructions without the need to use information from neighbouring elements. With this approach the global solution is not continuous but has jumps at the interfaces between the elements. At first glance, these methods may seem quite unsuitable for the discretization of elliptic equations because the discontinuous behavior of the numerical solution does not match well with the smoothness of the solution of an elliptic problem. However, the application of the DG framework to the solution of elliptic problems has proved to be remarkably successful. Several approaches have been proposed for the study of elliptic problems in the DG framework, see [3] for a comprehensive review. The first works related to these problems appeared

---

*Correspondence to: E-mail: andrea_ferrero@polito.it

several years ago. Lions [4] proposed the use of a penalization approach for the imposition of Dirichlet boundary conditions (BCs) in elliptic problems. His idea was to substitute the exact Dirichlet BC with a weak BC which reproduces the same effect of the original BC when a penalization factor tends to infinity. Nitsche [5] proposed an improved technique which was proven to be consistent. These ideas were subsequently adopted to impose inter-element continuity in the context of internal penalization methods (see e.g. the work of Babuska and Zlamal [6]) and were used to compute the diffusive fluxes exchanged between adjacent elements. Bassi and Rebay [7] proposed a DG method (BR1) for the discretization of the compressible Navier-Stokes equations which was subsequently generalized by Cockburn and Shu [8] with the Local Discontinuous Galerklet (LDG) method. Bassi, Rebay et al. [2] proposed also a second version of their scheme (BR2) in which a compact stencil is obtained thanks to the use of lifting operators. Similar motivations drove Peraire and Persson [10] to introduce the Compact Discontinuous Galerkin (CDG) which can be seen as a compact evolution of the LDG. A contribution was also given by Baumann and Oden [9] with a method based on a non-symmetric bilinear form. A different path was chosen by Gassner, Lorcher and Munz [11] who proposed a numerical flux for the diffusion terms in finite volume or DG schemes which is based on an exact solution of the diffusion equation with piecewise polynomial initial data.

In 2005 van Leer and Nomura [12] proposed the recovery method (RDG) in which diffusive fluxes are computed after a recovery procedure which gives a smooth solution that in the weak sense is indistinguishable from the piecewise continuous discrete solution. A similar approach was proposed by Dumbser [13] with the $P_N P_M$ scheme in which both convective and diffusive fluxes are computed through a reconstruction of order higher than the local element reconstruction. Luo et al. [14] presented a work in which they applied a reconstruction method for the Navier-Stokes equations, following the approach originally introduced by Dumbser et al. [15] in the context of Euler equations.

Other methods recently proposed exploit the idea of recovery. Huynh [16] performed a stability study for several of these schemes. He also proposed a 1D method which can be extended to 2D on quadrilateral meshes via tensor products. Borrel and Ryan [17] proposed the elastoplast method in which the recovery idea is applied to a small rectangle overlapping the interface between two elements. French et al. [18] introduced a modified recovery scheme which makes use of a penalty term. Moreover, in time several versions of the original recovery scheme were proposed by van Leer and co-workers. While the first version was based on a weak formulation twice partially integrated (RDG2x), a second version based on a weak formulation (RDG1x), requiring a single partial integration, was subsequently introduced. Lo and van Leer [1] suggested that this second version could be more suited to the discretization of the non linear Navier-Stokes equations, thanks to some simplifications in the switching to primitive variables. For this reason, the scheme proposed in this work will be compared mainly with the RDG1x scheme.

In this brief historical review it is possible to see some common lines which have driven the evolution of DG methods for elliptic problems. One of these is the compactness of the discretization. A compact stencil is very useful to simplify the implementation of codes on unstructured grids, and to improve scalability in parallel computations. Other important aspects are flexibility and robustness in dealing with highly distorted meshes and p-adaptive algorithms.

The aim of this work is to modify the original recovery method (RDG1x) focusing the attention on high order discretizations of convection-diffusion problems (Section 4). In particular, we propose a recovery basis which makes the scheme flexible and allows a robust implementation of p-adaptive algorithms. This basis has less terms than the original recovery basis and remains well conditioned also in the presence of higly distorted meshes. Furthermore the proposed discretization shows a larger stability limit than the RDG1x method and the implementation is simplified (Section 6). Moreover, a general procedure for the weak imposition of Dirichlet BCs in the framework of recovery-based methods (Section 5) is proposed and tested in the presence of curvilinear wall elements. The proposed boundary procedure is truly local and acts only on the boundary element. In contrast, the original procedure [19] proposed for the RDG1x method needs information also from the neighbouring elements.

The results obtained with the proposed discretization were compared not only with the original RGD1x method but also with the widely used BR2 method, which represents a reference in this field.

2. GOVERNING EQUATIONS

The numerical method described in this work will be applied for simplicity to 2D problems. The approach can be easily extended to 3D simulations. First of all, the 2D heat equation is considered:

$$\frac{\partial T}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = q \qquad (x,y) \in \Omega \subset R^2, t \in R_0^+ \tag{1}$$

where T is the temperature, q is a source term, and F and G are the fluxes. If we consider the Fourier law we obtain:

$$F = -\kappa \frac{\partial T}{\partial x} \qquad G = -\kappa \frac{\partial T}{\partial y} \tag{2}$$

Here κ denote the diffusivity coefficient which is assumed constant.
The heat equation will be used as a benchmark for the validation of the discretization of the diffusive fluxes.

The main purpose of this work is to integrate non linear convection-diffusion problems, such as compressible Navier-Stokes, which in 2D can be written as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{3}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla P + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) - \nabla \cdot \bar{\bar{\tau}} = 0 \tag{4}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot ((E + P)\mathbf{v}) - \nabla \cdot (\bar{\bar{\tau}} \cdot \mathbf{v}) - \nabla \cdot (\lambda \nabla T) = 0 \tag{5}$$

$$\bar{\bar{\tau}}_{ij} = \mu \left[\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j} \right) - \frac{2}{3}(\nabla \cdot \mathbf{v})\delta_{ij} \right] \tag{6}$$

where $\rho, \mathbf{v}$, P, E, τ, λ and μ are density, velocity vector, pressure, total energy per unit volume, viscous stresses, conductibility and dynamic viscosity, respectively. The set of equations is completed by the ideal gas law. Conductibility and dynamic viscosity are assumed constant.

3. NUMERICAL METHOD

The numerical scheme described in this work is based on the method of lines in which space and time are discretized separately. The space discretization is based on the discontinuous Galerkin approach, while the time discretization is performed with explicit Runge-Kutta algorithms or with the implicit Euler scheme .

3.1. Discontinuous Galerkin discretization in space

The space discretization will be described in the case of a general 2D governing equation for the variable u:

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = q \qquad (x,y) \in \Omega \subset R^2, t \in R_0^+ \tag{7}$$

Assume that the domain Ω is subdivided into a collection of non overlapping elements Ω_e. Consider a generic element Ω_e: let V_h be the finite dimensional functional space which will be used to approximate the solution inside Ω_e. Consider a proper basis for V_h which contains N_e basis

functions Φ_i. The numerical approximation of the solution (u_h) belongs to the space V_h and can be represented using a linear combination of the basis functions:

$$u_h(\mathbf{x}, t) = \tilde{\mathbf{u}} \cdot \mathbf{\Phi} = \sum_{i=1}^{N_e} \tilde{u}_i \Phi_i \tag{8}$$

If u_h is substituted in the governing equation (7) the residual R_h is obtained:

$$R_h = \frac{\partial u_h}{\partial t} + \frac{\partial F_h}{\partial x} + \frac{\partial G_h}{\partial y} - q \tag{9}$$

Galerkin's approach requires that the projection of the residual on the space V_h is set equal to zero:

$$\int_{\Omega_e} R_h \nu dx dy = \int_{\Omega_e} \frac{\partial u_h}{\partial t} \nu dx dy + \int_{\Omega_e} \frac{\partial F_h}{\partial x} \nu dx dy + \int_{\Omega_e} \frac{\partial G_h}{\partial y} \nu dx dy - \int_{\Omega_e} q \nu dx dy = 0 \quad \forall \nu \in V_h \tag{10}$$

The weak form of the discontinuous Galerkin discretization is derived by using integration by parts:

$$\int_{\Omega_e} \frac{\partial u_h}{\partial t} \nu dx dy + \int_{\partial\Omega_e} (\hat{F}_h n_x + \hat{G}_h n_y) \nu ds - \int_{\Omega_e} (\frac{\partial \nu}{\partial x} F_h + \frac{\partial \nu}{\partial y} G_h) dx dy - \int_{\Omega_e} q \nu dx dy = 0 \tag{11}$$

where n_x and n_y are the cartesian components of the outward-pointing unit normal vector and s is a curvilinear cohordinate along the element boundary. The last equation shows that the integral over the element boundary $\partial\Omega_e$ requires the knowledge of fluxes ($\hat{F}_h$ and $\hat{G}_h$) at the interface between neighbouring elements. These quantities are obtained introducing a numerical flux. The convective and diffusive contributions to this flux are evaluated separately. The first one is computed with an approximate Riemann solver, following the approach proposed by Osher and Solomon [21] and Pandolfi [22] for finite volume schemes. The numerical diffusive flux will be introduced in Section 4.

Eq. (11) has to be evaluated for all the N_e functions which define the basis of V_h, i.e. setting $\nu = \Phi_j$ with $j = 1, ..., N_e$. In this way the time evolution of the element's degrees of freedom is described by the following N_e equations:

$$\int_{\Omega_e} \sum_{i=1}^{N_e} \frac{\partial \tilde{u}_i}{\partial t} \Phi_i \Phi_j dx dy + \int_{\partial\Omega_e} (\hat{F}_h n_x + \hat{G}_h n_y) \Phi_j ds - \int_{\Omega_e} (\frac{\partial \Phi_j}{\partial x} F_h + \frac{\partial \Phi_j}{\partial y} G_h) dx dy +$$
$$- \int_{\Omega_e} q \Phi_j dx dy = 0 \quad 1 \le j \le N_e \tag{12}$$

The system can be rewritten in compact form through the introduction of the element mass matrix ($[\mathbf{M}]$) and the basis matrix ($\mathbf{\Phi}$),

$$[\mathbf{M}]_{ij} = \int_{\Omega_e} \Phi_i \Phi_j dx dy \tag{13}$$

$$[\mathbf{M}] \frac{\partial \tilde{\mathbf{u}}}{\partial t} = - \int_{\partial\Omega_e} (\hat{F}_h n_x + \hat{G}_h n_y) \mathbf{\Phi} ds + \int_{\Omega_e} (\frac{\partial \mathbf{\Phi}}{\partial x} F_h + \frac{\partial \mathbf{\Phi}}{\partial y} G_h) dx dy + \int_{\Omega_e} q \mathbf{\Phi} dx dy \tag{14}$$

3.2. Basis functions and mapping

The choice of the basis functions strongly affects the behavior of the numerical discretization. Indeed, the time evolution of the numerical solution depends on the linear system (14) in which the mass matrix appears. A bad choice of the basis functions can lead to an ill-conditioned mass matrix and this can deteriorate the accuracy of the numerical solution. For this reason we chose to use a hierarchical orthonormal basis obtained by the application of the modified Gram-Schmidt orthonormalization to a set of monomials defined in physical space. Furthermore, this

choice helps to reduce the computational cost with explicit time integration because, thanks to the orthonormalization, the element mass matrix is the identity and so it is not necessary to multiply the right hand side of (14) for the inverse of the mass matrix at each time step. In particular, we followed the approach of Bassi et al. [23]. They suggest to start the modified Gram-Schmidt procedure with a set of monomials defined in the physical space on a reference frame centered on the element's center of mass and aligned to its axis of inertia. They use this basis for general shaped elements obtained by agglomeration. In the present work this basis is employed for both triangles and quadrangles. In this way an element with a reconstruction of order p has $N_e = (p+1)(p+2)/2$ degrees of freedom, independently on its shape. The size of the particular element basis chosen for this work is not related to the recovery basis presented in Section 4.2. In fact other choices are possible, for example a tensor product basis (with $N_e = (p+1)^2$) can be efficiently used for quadrilateral elements.

Integrals are approximated through the use of Gauss quadrature formulas (from [24]) on the transformed reference element. The number of quadrature points is chosen in order to exactly integrate polynomials of order $2p$ on the reference element. The mapping between the general curvilinear quadrilateral element and the reference square are computed with the Serendipity mappings reported by Oñate [25]. Curvilinear triangular elements are transformed into the reference triangle with standard mappings also from [25].

3.3. Explicit and implicit time integration

In this work several explicit Runge-Kutta algorithms are employed. First order accuracy is obtained with the forward Euler scheme. We use two and three stages Total Variation Diminishing Runge-Kutta methods (RK2TVD and RK3TVD), proposed by Gottlieb and Shu [26] in order to achieve second and third-order accuracy. Fourth-order time accurate computations are performed using the low storage five stages Runge-Kutta method (RK4LS) described in [27]. In the general case in which convection-diffusion equations are considered the diffusion and convection stability limits are evaluated separately:

$$\Delta t_d = \sigma \frac{H_e^2}{\kappa(2p_e + 1)} \tag{15}$$

$$\Delta t_c = \sigma \frac{H_e}{c(2p_e + 1)} \tag{16}$$

where σ, c, H_e and p_e are the stability coefficient, maximum signal speed and element characteristic size and order. The allowed time step in the generic element is chosen as the minimum between the diffusion and convection step:

$$\Delta t = \min(\Delta t_d, \Delta t_c) \tag{17}$$

Implicit time integration is also considered in this work. In particular, the first order backward Euler method is used for the test of Section 7.4. The jacobian matrix required by this approach is evaluated analytically.

4. A RECOVERY-BASED APPROACH

4.1. The recovery idea

The recovery method (RDG) proposed by van Leer and Nomura [12] is an elegant way to obtain the numerical diffusive fluxes.

Consider the interface between two elements Ω_A and Ω_B. The goal is to recover a smooth differentiable function (u^r) on the union of these two elements ($\Omega_C = \Omega_A \cup \Omega_B$) in order to compute gradients on the quadrature points along the interface. The key idea is that the recovered function has to be equivalent in the weak sense to the original solution on the two elements, namely

$$\int_{\Omega_A} u^r \Phi_n dx dy = \int_{\Omega_A} u_h \Phi_n dx dy \qquad 1 \le n \le N_A$$

$$\int_{\Omega_B} u^r \Phi_n dx dy = \int_{\Omega_B} u_h \Phi_n dx dy \qquad 1 \le n \le N_B \tag{18}$$

A set of basis functions $\boldsymbol{\Psi}$ with N_r degrees of freedom can be introduced for the recovered solution:

$$u^r = \tilde{\mathbf{u}}^{\mathbf{r}} \cdot \boldsymbol{\Psi} = \sum_{j=1}^{N_r} \tilde{u}_j^r(t) \Psi_j(x, y) \tag{19}$$

Substituting (19) and (8) into (18) the following two sets of linear equations are obtained:

$$[\mathbf{R}_A] \tilde{\mathbf{u}}^{\mathbf{r}} = [\mathbf{M}_A] \tilde{\mathbf{u}}_A$$
$$[\mathbf{R}_B] \tilde{\mathbf{u}}^{\mathbf{r}} = [\mathbf{M}_B] \tilde{\mathbf{u}}_B \tag{20}$$

in which $[\mathbf{M}_A]$, $[\mathbf{M}_B]$, $\tilde{\mathbf{u}}_A$ and $\tilde{\mathbf{u}}_B$ are the mass matrices and the vectors of the degrees of freedom of the two elements A and B, respectively. The recovery matrices $[\mathbf{R}_A]$ and $[\mathbf{R}_B]$ are defined as

$$[\mathbf{R}_A]_{nj} = \int_{\Omega_A} \Psi_j \Phi_n dx dy \qquad 1 \le n \le N_A, \quad 1 \le j \le N_r$$

$$[\mathbf{R}_B]_{nj} = \int_{\Omega_B} \Psi_j \Phi_n dx dy \qquad 1 \le n \le N_B, \quad 1 \le j \le N_r. \tag{21}$$

Introducing the interface recovery matrix $[\mathbf{R}]$, the set of equations (20) can be expressed in a more compact form:

$$\begin{bmatrix} [\mathbf{R}_A] \\ \hline [\mathbf{R}_B] \end{bmatrix} \tilde{\mathbf{u}}^{\mathbf{r}} = [\mathbf{R}] \tilde{\mathbf{u}}^{\mathbf{r}} = \begin{bmatrix} [\mathbf{M}_A] & 0 \\ 0 & [\mathbf{M}_B] \end{bmatrix} \left\{ \begin{array}{c} \tilde{\mathbf{u}}^{\mathbf{A}} \\ \hline \tilde{\mathbf{u}}^{\mathbf{B}} \end{array} \right\}. \tag{22}$$

If an orthonormal basis is used, the mass matrix within each element is the identity, and so the matrix multiplication on the right hand side can be removed,

$$[\mathbf{R}] \tilde{\mathbf{u}}^{\mathbf{r}} = \left\{ \begin{array}{c} \tilde{\mathbf{u}}^{\mathbf{A}} \\ \hline \tilde{\mathbf{u}}^{\mathbf{B}} \end{array} \right\}. \tag{23}$$

From the previous derivation it can be seen that the recovery matrix $[\mathbf{R}]$ has dimension $(N_A + N_B) \times N_r$. The most difficult task in the recovery approach is the choice of the recovery basis $\boldsymbol{\Psi}$.

Van Leer and Lo [19] considered the case in which $N_A = N_B$, and they set $N_r = (N_A + N_B)$. They introduced a provisional recovery basis $\boldsymbol{\Psi}$ which makes the recovery matrix non singular. The elegance of their idea is related to the next step. They used the columns of the inverse recovery matrix to define a final recovery basis $\hat{\boldsymbol{\Psi}}$ which is in the weak sense indistinguishable from the elements discontinuous basis (van Leer and Lo [28]). In this way the degrees of freedom of the two elements which share an interface can be used to put in evidence both the discontinuous hyperbolic nature or the continuous parabolic nature of the underlying equations simply by switching from a basis ($\boldsymbol{\Phi}$) to another ($\hat{\boldsymbol{\Psi}}$).

$$u^r = \sum_{j=1}^{N_A} \tilde{u}_j^A \hat{\Psi}_j + \sum_{j=1}^{N_B} \tilde{u}_j^B \hat{\Psi}_{N_A+j}. \tag{24}$$

4.2. Enhanced Stability Recovery (ESR) method

The approach proposed by van Leer, Nomura and van Raalte [19] makes it possible to enhance the accuracy of the viscous fluxes discretization because the recovered solution has twice the number of degrees of freedom with respect to the element solution. This advantage is important in pure diffusion problems but can be lost in convection-diffusion problems if the convective fluxes are computed using the element internal reconstruction. There are some approaches in which the information from the neighbouring elements is used to improve also the accuracy of the convective terms, see for example the PNPM method [13] or the approach proposed by Nourgaliev et al. [29].

We believe that one of the main reasons for the current success of DG methods is the possibility of obtaining high order reconstructions without the need of data from neighbouring elements. For this reason we prefer to adopt a classical approach, in which the convective fluxes are computed using the internal element reconstruction. As a consequence the effects of a higher order approximation of the diffusion terms are mitigated by the lower order approximation of the convective terms.

Furthermore, in the original recovery approach it is necessary to choose a provisional recovery basis (Ψ) which makes the recovery matrix ($[\mathbf{R}]$) square and invertible. This is not a trivial task in multidimensional problems, especially when high order reconstructions are used on irregular unstructured meshes. For example, van Leer et al. [30] reported that the RDG1x scheme becomes unstable for $p > 2$, if the gradient in the volume integral is computed with one of the approaches described in [1]. In contrast, the discretization proposed in this work was successfully tested up to $p = 6$ (see Section 7.3).

Van Leer, Nomura and van Raalte [19] chose to use an anisotropic monomial recovery basis in which the reconstruction orders in the direction normal and tangential to the interface are $(2p + 1)$ and p, respectively. Even if this approach improves the order of the scheme on cartesian meshes, a standard $(p + 1)$ convergence order is obtained on general unstructured meshes [28].

One of the final goals of this work is the discretization of convection-diffusion problems in which p-adaptivity strategies can bring important benefits. In this context, we observe that it is not clear how to extend the RDG1x approach (Eq. 24 which requires $N_r = N_A + N_B$) to problems in which the reconstruction order changes from one element to another ($N_A \neq N_B$). In the literature there are no applications of the original recovery approach with p-adaptive algorithms to the best of our knowledge.

Thus we choose an alternative approach in order to obtain a general and flexible scheme (Enhanced Stability Recovery (ESR)), which can be easily adopted in the context of p-adaptive algorithms. In particular we give up the constraint of a square recovery matrix (i.e. we set $N_r < N_A + N_B$) and we solve the over-determined system (23) with the least square method:

$$[\mathbf{R}]^T [\mathbf{R}] \tilde{\mathbf{u}}^{\mathbf{r}} = [\mathbf{R}]^T \left\{ \frac{\tilde{\mathbf{u}}^{\mathbf{A}}}{\tilde{\mathbf{u}}^{\mathbf{B}}} \right\} \tag{25}$$

Thanks to a proper choice of the recovery basis (defined in the following) the least square problem is well conditioned and the matrix $[\mathbf{R}]^T [\mathbf{R}]$ is invertible:

$$\tilde{\mathbf{u}}^{\mathbf{r}} = \left([\mathbf{R}]^T [\mathbf{R}] \right)^{-1} [\mathbf{R}]^T \left\{ \frac{\tilde{\mathbf{u}}^{\mathbf{A}}}{\tilde{\mathbf{u}}^{\mathbf{B}}} \right\} = [\tilde{\mathbf{R}}] \left\{ \frac{\tilde{\mathbf{u}}^{\mathbf{A}}}{\tilde{\mathbf{u}}^{\mathbf{B}}} \right\} \tag{26}$$

In this way it is sufficient to perform a matrix-vector product for each interface and the recovery coefficients are obtained. Then these coefficients can be multiplied by the gradient of the recovery basis functions in order to obtain the diffusive fluxes at the interface quadrature points (first term on the right hand side of Eq. (14)). If the domain is fixed, the rectangular recovery matrix ($[\tilde{\mathbf{R}}]$) can be precomputed since it depends only on the mesh. In Section 6 more details on the implementation are given.

In the proposed approach, the recovered solution is used only for the evaluation of the diffusive fluxes at the interface. Indeed, the diffusive fluxes within the volume integral (second term on the right hand side of Eq. (14)) are computed inside each element, and therefore are given by the product between the element degrees of freedom and the gradient of the element basis functions.

The most important aspect of the method is the choice of the recovery basis Ψ. We use an orthonormal basis obtained by the modified Gram-Schmidt procedure. The orthonormalization makes use of the L^2 inner product $(f, g) = \int_{\Omega_A \cup \Omega_B} f g \, dx dy$ in which the integral is evaluated on the union of the two elements. This approach makes it possible to ensure that the condition number of the recovery system remains low. We propose to initialize the orthonormalization procedure with a particular starting monomial basis defined according to rules, depending on the degree p_A and p_B of the polynomials in the two elements A and B, defining the interface. Three cases will be described depending on the order of the two elements. Consider a reference frame centered on the mid-point

of the interface and with the axis aligned to the normal and tangential directions (ξ, η) as shown in Figure 1:

Figure 1. Interface reference frame for constructing the recovery basis.

- $p_A = p_B = p$
  First of all we consider the case in which the two elements have the same order p. The starting recovery basis for the initialization of the orthonormalization procedure is obtained by considering the monomial basis of order p in the variables (ξ, η) and adding all the monomials with order $p + 1$ in which ξ appears. In other words these additive functions consist in the first $p + 1$ monomials of the $(p + 1)th$ row of the Pascal triangle. We emphasize that it is these additive functions that make the difference. Indeed, our preliminary experiments showed that if the additive functions are not used or if only the term ξ^{p+1} is included, the convergence rate of the scheme can be compromised. Since the order of the recovery reconstruction is reduced with respect to the original recovery approach, it is easier to control the condition number of the recovery matrix. Moreover, this choice is related to the improvement in the stability limit observed in the numerical experiments presented below.
  An example of the proposed starting basis in the case $p = 2$ is shown in Figure 2. For the sake of comparison, Figure 3 reports the provisional basis for $p = 2$ obtained according to the original recovery approach.

Figure 2. Preliminary recovery basis before orthonormalization for the case $p_A = p_B = 2$, in the Enhanced Recovery approach.

Figure 3. Preliminary recovery basis with the original recovery approach for the case $p_A = p_B = 2$.

- $|p_A - p_B| = 1$
  The starting recovery basis is obtained by considering the monomial basis of order $\tilde{p} = \max(p_A, p_B)$ and adding all the monomials of order $\tilde{p} + 1$ in which ξ appears. It is possible to use the maximum order between p_A and p_B if there is enough information in the union of the two elements to obtain a recovered solution with this accuracy, i.e. $N_r \leq N_A + N_B$. This condition is satisfied if $min(p_A, p_B) \geq 1$. Thus, the p-adaptive strategy avoids the use of elements with $p = 0$.

- $|p_A - p_B| > 1$
  It could happen that a p-adaptive algorithm introduces a jump of two or more orders between neighbouring elements. In this case it is impossible to choose the recovery basis with $\tilde{p} = max(p_A, p_B)$. Indeed, in the case $|p_A - p_B| > 1$, the recovery degrees of freedom would be more than the sum of the degrees of freedom in the two elements if $\tilde{p} = max(p_A, p_B)$. For this reason, we locally reduce the order of the recovery, choosing $\tilde{p} = max(p_A, p_B) - 1$, i.e. at the interface between elements with a different accuracy, the algorithm chooses an intermediate value for the degree of the polynomial being reconstructed. This reflects the fact that the adaptive algorithm assigns a jump in the accuracy across the interface, so it seems reasonable to choose an intermediate state at the interface. Thus, in this case, we construct the starting basis with a monomial basis of order $\tilde{p} = \max(p_A, p_B) - 1$ and we add all the monomials of order $\tilde{p} + 1$ in which ξ appears. In our tests we found that the presence of strong jumps in the order distribution across the domain should be prevented whenever possible (see Section 8.1). For this reason, we try to avoid the condition $|p_A - p_B| > 1$ in the simulations by choosing a more regular order distribution. However, if the condition is verified the algorithm is able to manage this case, as described above.

Therefore, the starting recovery basis has always $N_r = (\tilde{p}+1)(\tilde{p}+2)/2 + (\tilde{p}+1)$ degrees of freedom, where $\tilde{p}$ is equal to p if a uniform order distribution is used. Otherwise $\tilde{p}$ can be set equal to $max(p_A, p_B)$ or $max(p_A, p_B) - 1$, depending on the jump in the order.

As far as numerical quadrature is concerned, the integral on the interface is computed with a suitable quadrature formula in order to satisfy the accuracy requirements of both elements. In particular, if the two elements require M_A and M_B points for the integrals on their edges, the formula with $M = max(M_A, M_B)$ is adopted.

Finally, a comment on the use of piecewise constant reconstructions. When $p_A = p_B = 0$ the recovery reconstruction has only two degrees of freedom: a constant term and a linear term related to the direction normal to the interface. From this information, only the normal derivative can be obtained from the recovery solution. If the heat equation is considered this information is enough to compute the fluxes across the interface. On the other hand, if a more complex system of equations is considered (i.e. Navier-Stokes), not only the normal derivative but also the tangential derivative have to be computed in order to evaluate the fluxes. For this reason, we prevent the presence of elements with $p = 0$ in the domain. Note that the DG method is usually chosen for its high accuracy. Thus, the occurrence of piecewise constant reconstructions is expected to be unlikely in most applications.

We emphasize the importance of using the Gram-Schmidt orthonormalization not only on the element basis but also on the recovery basis. This approach makes the method robust and able to deal with highly distorted meshes. For example we report in Table I the effect of the recovery basis orthonormalization on the 2-norm condition number of the matrix $\left([\mathbf{R}]^T[\mathbf{R}]\right)$ which has to be inverted in Eq. 26. The data refer to the interface between the two distorted elements shown in Figure 4 for the case $p_A = p_B$. Orthonormalization is applied on the element basis functions in both cases, but only in the second case it is also applied to the recovery basis. These results suggest that the use of orthonormalization on the proposed recovery basis becomes fundamental when high order reconstructions are employed on irregular meshes. The condition number of the matrix was estimated with the free software package GNU Octave [31].

It can be expected that the use of the orthonormalization procedure on the recovery basis can be useful also for other recovery-based schemes, since they all involve the definition of a recovery procedure which is influenced by the behavior of the recovery basis.

Figure 4. Interface between two distorted elements for the test of Table I.

4.3. Extension to 3D

In this Section some considerations on the extension of the method to 3D are reported. In particular it is necessary to define how to build the recovery basis in 3D. Consider a reference frame with the origin in the geometric center of the interface. Its axis are aligned with the normal and tangential directions (ξ, η, ζ).

First of all, the case $p_A = p_B = \tilde{p}$ is studied. The starting recovery basis can be obtained by considering the monomial basis of order $\tilde{p}$ and adding all the monomials with order $\tilde{p}+1$ in which ξ appears. With this approach the size of the recovery basis is $N_r = (\tilde{p}+1)(\tilde{p}+2)(\tilde{p}+3)/6 + (\tilde{p}+1)(\tilde{p}+2)/2$.

The cases $|p_A - p_B| = 1$ and $|p_A - p_B| > 1$ can be managed in a way similar to the 2D approach. The same problems described for $p = 0$ in the 2D context are present also in 3D.

Another important issue which can become important in 3D is the memory requirement. In Section 6 there are suggestions on implementation details which can help in saving memory. Future work will be devoted to the assessment of the method in 3D.

5. BOUNDARY CONDITIONS

In this Section we describe how boundary conditions for diffusive equations should be imposed with the Enhanced Recovery Method. Neumann BCs are simply satisfied by setting the exact value of the fluxes on the boundary integral. Dirichlet BCs require further considerations.

In the context of discontinuous Galerkin methods it is possible to impose Dirichlet BCs in both strong or weak form [32]. The strong approach consists in forcing the solution reconstruction to exactly satisfy a prescribed condition on the boundary. The weak approach is based on the idea that the error in the enforcement of the BC's and the error in the interior solution can be of the same size. In other words, the accuracy required in the interior of the domain is the same as the accuracy imposed at the boundary: in this approach therefore there is an error between the value prescribed and the solution reconstructed on the boundary, but this error vanishes in the limit of an infinite mesh refinement, with the same rate of the error in the interior of the domain.

From a practical point of view, it can be shown that the use of strong Dirichlet BCs can introduce stability limitations and spurious oscillations in the numerical solution (see for example [33]). Furthermore several discretization methods allow the weak imposition of Dirichlet BCs in a natural way. For example, in the BR2 scheme this is simply done through the definition of a proper ghost state in all the boundary quadrature points. The extension to this approach in recovery-based schemes could be non trivial. Indeed, the recovery procedure is applied to the union of two elements and so a ghost element would be required in this case, not just a local ghost state. The definition of the proper values in the ghost element can be obtained through symmetry considerations between the boundary element and the ghost element. For example Borrel and Ryan use this approach in their paper [17] on the elastoplast method. The problem is that this procedure may become ambiguous in the presence of curvilinear elements, because in this case the boundary curve is not a symmetry axis between the boundary element and the ghost element.
Another path was followed by van Leer, Nomura and van Raalte [19]. They proposed to impose the BCs on the recovery solution. They substitute the missing information in the boundary element with data from one or more interior elements. They observed that this procedure may reduce the stability range of the method. Furthermore this approach is not trivial when unstructured meshes are used, because there are multiple possible choices.

Two of the main reasons for the success of DG methods are the compactness of the reconstruction and the simplification in the treatment of BC's when high order reconstructions are employed. In order to keep these positive features, we developed an alternative approach which makes it possible to impose both strong or weak Dirichlet BC's in recovery-based DG methods, without the need to use ghost elements or data from other interior elements.

5.1. Dirichlet BC's in strong form

In our approach boundary conditions are imposed on the element internal reconstruction, not on the recovery solution.
The strong imposition of Dirichlet BC's requires that the numerical solution at the quadrature points of the boundary elements is equal to the prescribed values. In other words, the degrees of freedom inside the element have to be altered in order to satisfy the BC's. Consider for example a discontinuous Galerkin scheme with N_e degrees of freedom inside the element and M quadrature points on the edge ($M < N_e$). The imposition of the BC's on the quadrature points gives M equations for each element on the boundary:

$$u_w^m = u_w(\mathbf{x}_q^m) = \sum_{i=1}^{N_e} (\tilde{u}_i + \Delta \tilde{u}_i)\Phi_i(\mathbf{x}_q^m) \qquad 1 \leq m \leq M \qquad (27)$$

where u_w^m represents the prescribed wall value in the quadrature point m.
If the procedure is forced to be conservative the mean value has to be preserved ($\Delta \tilde{u}_1 = 0$). In this way the system (27) has $N_e - 1$ unknowns which describe the perturbation of the elements degrees of freedom ($\Delta \tilde{u}_i, 2 \leq i \leq N_e$). In general the set of equations (27) is under-determined and so more

information has to be imposed. In particular we chose to minimize the L^2-norm of the perturbation $\Delta\tilde{\mathbf{u}}$:

$$min(|\Delta\tilde{\mathbf{u}}|_2) = min(\sum_{i=2}^{N_e} \Delta\tilde{u}_i^2). \tag{28}$$

The minimization problem can be solved using a Lagrange multiplier, μ. In this approach, equation (28) is the goal and equations (27) are the constraints:

$$\frac{\partial}{\partial\Delta\tilde{u}_i} \left[\sum_{i=2}^{N_e} \Delta\tilde{u}_i^2 + \sum_{m=1}^{M} \mu_m \left(u_w^m - \sum_{i=1}^{N_e} (\tilde{u}_i + \Delta\tilde{u}_i)\Phi_i(\mathbf{x}_q^m) \right) \right] = 0. \tag{29}$$

In other words we look for the smallest perturbation which is able to satisfy the BC's. The set of equations (29) and (27) defines the linear system (30) in which there are $(N_e - 1 + M)$ equations and $(N_e - 1 + M)$ unknowns: the perturbations of the element degrees of freedom (with the exception of $\Delta\tilde{u}_1$ which is set to zero and is already known) plus the Lagrange multipliers. The boundary matrix $[\mathbf{B}]$ and the boundary vector $\{\mathbf{W}\}$ are explicitly written in (31) and (32) :

$$[\mathbf{B}] \begin{Bmatrix} \Delta u_2 \\ \vdots \\ \Delta u_{N_e} \\ \mu_1 \\ \vdots \\ \mu_M \end{Bmatrix} = \{\mathbf{W}\} \tag{30}$$

$$[\mathbf{B}] = \begin{bmatrix} 2 & 0 & \cdots & 0 & -\Phi_2(\mathbf{x}_q^1) & \cdots & -\Phi_2(\mathbf{x}_q^M) \\ 0 & 2 & 0 & \vdots & \vdots & \cdots & \vdots \\ \vdots & 0 & \ddots & 0 & \vdots & \cdots & \vdots \\ 0 & \cdots & 0 & 2 & -\Phi_{N_e}(\mathbf{x}_q^1) & \cdots & -\Phi_{N_e}(\mathbf{x}_q^M) \\ \Phi_2(\mathbf{x}_q^1) & \cdots & \cdots & \Phi_{N_e}(\mathbf{x}_q^1) & 0 & \cdots & 0 \\ \vdots & \cdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \Phi_2(\mathbf{x}_q^M) & \cdots & \cdots & \Phi_{N_e}(\mathbf{x}_q^M) & 0 & \cdots & 0 \end{bmatrix} \tag{31}$$

$$\{\mathbf{W}\} = \begin{Bmatrix} 0 \\ \vdots \\ 0 \\ u_{w1} - \sum_{i=1}^{N_e} \tilde{u}_i\Phi_i(\mathbf{x}_q^1) \\ \vdots \\ u_{wM} - \sum_{i=1}^{N_e} \tilde{u}_i\Phi_i(\mathbf{x}_q^M) \end{Bmatrix}. \tag{32}$$

The solution of this system gives the perturbations which have to be applied to the element. In particular, the Lagrange multipliers are not needed and they can be eliminated. Thus it is possible to define a reduced boundary matrix $[\hat{\mathbf{B}}]$ such that:

$$\begin{Bmatrix} \Delta u_2 \\ \vdots \\ \Delta u_{N_e} \end{Bmatrix} = [\hat{\mathbf{B}}] \begin{Bmatrix} u_w^1 - \sum_{i=1}^{N_e} \tilde{u}_i\Phi_i(\mathbf{x}_q^1) \\ \vdots \\ u_w^M - \sum_{i=1}^{N_e} \tilde{u}_i\Phi_i(\mathbf{x}_q^M) \end{Bmatrix} = [\hat{\mathbf{B}}] \{\hat{\mathbf{W}}\}. \tag{33}$$

The reduced boundary matrix $[\hat{\mathbf{B}}]$ can be obtained by the first $N_e - 1$ rows and the last M columns of the inverse of $[\mathbf{B}]$. At this point the gradient required by the boundary fluxes can be computed with the product between the perturbed degrees of freedom and the gradient of the element basis functions.

5.2. Dirichlet BCs in weak form

In the previous paragraph a procedure for the strong imposition of Dirichlet BC's has been presented. The element's degrees of freedom are perturbed in such a way that the element internal reconstruction exactly satisfies the condition on the boundary. After that, the gradient of the element solution is computed at the boundary quadrature points, yielding the boundary fluxes.

We adopt a similar procedure also for the weak imposition of the BC's. The only difference is that, after the fluxes computation, we remove the perturbation previously applied to the element. In this way the BC's are weakly imposed in the sense that they determine the value of the boundary fluxes, but there is not a direct alteration of the element's degrees of freedom. The procedure can be summarized as follows

- Compute the element's perturbation $\Delta \tilde{\mathbf{u}}$ with Eq. (33).
- Apply the perturbations to $\tilde{\mathbf{u}}$, $\tilde{\mathbf{u}} := \tilde{\mathbf{u}} + \Delta \tilde{\mathbf{u}}$.
- Compute the boundary gradients $\nabla u = \sum_{i=1}^{N_e} \tilde{u}_i \nabla \Phi_i$.
- Compute the boundary fluxes $\hat{F}_h(\nabla u)$
- Remove the perturbations $\tilde{\mathbf{u}} := \tilde{\mathbf{u}} - \Delta \tilde{\mathbf{u}}$.

All numerical results obtained with the ESR method which are presented in the following were computed with this weak enforcement of BC's.

6. IMPLEMENTATION AND COMPUTATIONAL COST

In this Section the implementation of the method is briefly described in order to identify the main contributions to the computational cost. In particular some considerations are made in order to compare the proposed approach with the RDG1x recovery method. Both schemes are compact in the sense that the computation of viscous fluxes requires the knowledge of information only from the first neighbours of the element. For simplicity a scalar 2D problem in which all the elements have the same order (p) is considered. In this case $N_e = (p+1)(p+2)/2$ and $N_r = (p+1)(p+2)/2 + (p+1)$. The domain is supposed to be fixed, so that the mesh does not change in time. The cost required by the computation of the integrals appearing in Eq. (14) can be decomposed in different contributions, analized below.

6.1. Interface integral computational cost

We start from the computational cost due to the evaluation of the diffusive fluxes on an interface with M quadrature points.
If the ESR method is used the following operations are required for each interface quadrature point:

$$\nabla u = [\mathbf{D}_r] \tilde{\mathbf{u}}^{\mathbf{r}} = [\mathbf{D}_r][\tilde{\mathbf{R}}] \left\{ \frac{\tilde{\mathbf{u}}^{\mathbf{A}}}{\tilde{\mathbf{u}}^{\mathbf{B}}} \right\} = [\mathbf{D}_r \tilde{\mathbf{R}}] \left\{ \frac{\tilde{\mathbf{u}}^{\mathbf{A}}}{\tilde{\mathbf{u}}^{\mathbf{B}}} \right\}.$$

The matrix with the derivatives of the recovery basis functions $[\mathbf{D}_r]$ and the recovery matrix $[\tilde{\mathbf{R}}]$ can be computed and multiplied at the beginning of the simulation since they depend only on the mesh. The cost for each interface is proportional to $M \times 2 \times 2N_e$.

In the RDG1x scheme the recovery coefficients $\tilde{\mathbf{u}}^{\mathbf{r}}$ coincide with the degrees of freedom inside the two elements and so the gradient can be obtained as

$$\nabla u = [\mathbf{D}_r] \left\{ \frac{\tilde{\mathbf{u}}^{\mathbf{A}}}{\tilde{\mathbf{u}}^{\mathbf{B}}} \right\}.$$

Thus, the cost for each interface is proportional to $M \times 2 \times 2N_e$ also in this case.

6.2. Volume integral computational cost

The computational cost of the volume integral will be evaluated assuming that L quadrature points are used.

The cost for the ESR method is related to the following operations:

- $\nabla u = [\mathbf{D}]\tilde{\mathbf{u}}$
  In each quadrature point the fluxes are computed using the internal element solution. The cost is proportional to $L \times 2 \times N_e$. The matrix with the derivatives of the element basis functions $[\mathbf{D}]$ can be precomputed since it depends only on the mesh.

The cost for the RDG1x method is related to the following operations:

- $\nabla u = \frac{1}{N_F} \sum_{i=1}^{N_F} [\mathbf{D}_r]_i \tilde{\mathbf{u}}_i^{\mathbf{r}}$
  In each quadrature point the gradient is computed as an average between all the gradients obtained by the several available recovery solutions (one for each face of the element). In this case the cost is proportional to $L \times N_F \times 2 \times 2N_e$ where N_F is the number of faces of the element. Lo and van Leer [1] proposed also another approach which improves accuracy but is more involved. In practice they use the recovery solution only for evaluating the derivative in the coordinate normal to the interface and, since there are more directions than dimensions, they introduce a least square approach. This implies that, in each internal quadrature point, a least square problem has to be solved. The cost in this case is greater than in the previous approach and the implementation is significantly more involved.

The previous considerations suggest that the main difference between the cost of the ESR method and the RDG1x method is related to the computation of the volume integral. Indeed, in the described implementation the cost per interface is the same for the two methods even if they use recovery basis with a different size. However, the RDG1x method requires more work than the ESR method in the volume integral, because it uses all the several recovery functions defined in the element (one for each face). In contrast, the ESR method is based on a very simple evaluation of the volume integral since it directly uses the gradient of the element solution. From a practical point of view, the CPU time per iteration required by the two methods might be similar but it is strongly influenced by the implementation (cache effects can significantly alter the speed) and by possible memory limitations which can forbid the storage of precomputed matrices.

Since the test cases considered in this work are 2D, there are no memory limitations, and all quantities that can be precomputed will be stored in memory at the beginning of each computation. However, memory requirements can become an important issue in 3D problems. In order to reduce memory usage it is useful to avoid the storage of the interface recovery matrix ($[\tilde{\mathbf{R}}]$). Indeed, it is more convenient to pre-compute and store the product between the matrix of the derivatives of the recovery basis functions ($[\mathbf{D}_r]$) and the recovery matrix ($[\tilde{\mathbf{R}}]$), for each quadrature point.

In the example described in this Section a scalar problem is studied. If a system of equations is considered (i.e. Navier-Stokes) the recovery approach gives the gradients of the conservative variables. In order to compute the fluxes it is necessary to switch to the gradients of the primitive variables. This can be done by computing the values of the conservative recovered variables at the interface. In this case too, it is possible to save memory, avoiding the storage of the recovery matrix and storing the product between the matrix containing the recovery functions, evaluated at the interface, and the recovery matrix.

The previous analysis considers only a problem with a fixed grid. If deforming meshes are employed, the recovery functions and the related matrices have to be re-computed during the simulation. This additional cost can be significant for both the RDG1x and the ESR methods. In this kind of problems recovery-based methods might be less efficient than other existing approaches, like, for example, the BR2 method.

7. NUMERICAL EXPERIMENTS

7.1. 1D Poisson equation

In order to evaluate the behavior of the proposed discretization we studied a 1D Poisson problem with a time-marching approach. The problem has the following exact solution which also defines Dirichlet BC's:

$$u(x) = 1 - x + \sin(2\pi x) \qquad 0 \leq x \leq 1. \tag{34}$$

Van Leer et al. [30] reported a comparison between several schemes (with $p = 1$ and forward Euler) for this test case. In Table II we report the dimensionless time step $\nu \Delta t / \Delta x^2$ obtained for the ESR method, comparing it with the stability limit for the BR2 and then RDG1x methods from [30]. Table III shows the L^2 error on the temperature and the convergence order obtained in our implementation for RDG1x, BR2 ($\eta_{BR2} = 1$) and the proposed ESR scheme. All results shown in the table, were obtained with $p = 1$ and forward Euler time integration. We took care of employing the BC's procedures originally proposed by the different authors. In particular for the RDG1x scheme, BC's are imposed on the recovery solution, exploiting information from the boundary and the first two elements close to it, as in [19]. For the BR2 scheme we defined a proper ghost state computed as a function of the BC's. For the ESR method we used the weak minimization procedure proposed in Section 5.2. Our numerical experiments show that the ESR method has a significantly larger stability limit than the other schemes. On the other hand RDG1x gives excellent results and is fourth order accurate ($2(p+1)$) while the other two schemes are second order accurate ($p+1$). This advantage of the RDG1x method is lost on multidimensional unstructured meshes in which its convergence order reduces to $p+1$ [19]. Even if the ESR method obtains the expected convergence order, it can be seen that its error is 60% larger than the error obtained by BR2 when $p = 1$. On the other hand, the time step for ESR can be chosen 65% larger than it is possible in BR2. From this first comparison, it can be seen that in pure diffusion problems with cartesian meshes the RDG1x method is the most accurate scheme between the three tested.

7.2. 1D Convection Diffusion problem

The final goal of this work is the application of the proposed discretization in convection-diffusion problems. For this reason we studied the following 1D model problem with a time marching approach:

$$\frac{\partial u}{\partial t} + 10 \frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} = 0 \qquad 0 \leq x \leq 1 \qquad u(0) = 1, u(1) = 0. \tag{35}$$

Simulations were carried out until the residual reached machine precision. Then, the converged numerical solution was compared with the exact steady state solution

$$u(x) = \frac{\exp(10x) - \exp(10)}{1 - \exp(10)}. \tag{36}$$

We performed computations with RDG1x, BR2 and ESR methods ($p = 1$) and the first order forward Euler time integration scheme. Three non uniform meshes with 5, 10 and 20 elements were used. The meshes are clustered in order to reproduce the aspect of the typical grids used in Navier-Stokes boundary layer simulations. The size of the element near the boundary $x = 1$ is $\Delta x = 0.08, 0.04$ and 0.02 for the different meshes. In Table IV the L^2-errors are reported for the three schemes. While in the previous test case the RDG1x scheme gave errors which are several orders of magnitude lower than the other schemes, now the errors for the different methods are significantly closer. The ratio between the error of the ESR and the RDG1x methods is equal to approximately 3-4 for the different meshes. Furthermore the stability limit of the discretization is dominated by the diffusive terms and so the ESR method can benefit from its double time step with respect to the RDG1x (the data in Table II still hold). These results are limited to a scalar linear problem with $p = 1$ but they suggest that when convective effects appear in the equations, the global errors obtained by the ESR and the RDG1x methods become closer because the bottle neck is represented by the convective discretization. We can expect that this difference becomes even smaller in high Reynolds number

industrial flows, in which the convective terms are particularly strong and the meshes are often highly distorted.

7.3. 2D Heat equation

In this Section the behavior of the proposed scheme is evaluated in 2D for several accuracies, including some very high order schemes. The linear 2D heat equation was solved in a square domain $(0, 1) \times (0, 1)$ with unstructured triangular meshes. The meshes were obtained starting from uniform cartesian grids in which quadrilaterals were splitted in triangles. The following Dirichlet BC's were prescribed:

$$\begin{cases} \quad\quad\quad T = 0 & \text{on } x = 1, y = 0 \text{ and } y = 1 \\ T = \sin(k\pi y) & \text{on } x = 0. \end{cases} \tag{37}$$

For the ESR method, BC's were imposed with the weak approach, based on the minimization procedure of Section 5.2. The parameter k is an integer which represents the number of semi-waves in the y-direction. In this test it was set to $k = 1$. The numerical solution was initialized with a uniform field $T_0 = 1$ and computations were carried out until the residuals reached machine precision. The L^2 norm of the temperature error was computed through a comparison between the obtained steady state results and the exact analytical solution

$$T = \frac{\sinh(k\pi(1-x))\sin(k\pi y)}{\sinh(k\pi)}. \tag{38}$$

The test was also performed with the widely used BR2 scheme in order to compare the behavior of the two methods. The stabilization parameter associated with this last scheme was set to $\eta_{BR2} = 4$ ($\eta_{BR2} > N_F$ according to [23]). The BC's for the BR2 scheme are weakly imposed by means of proper ghost values on the boundary quadrature points.

The mesh refinement analysis is reported in Table V. It can be seen that even with few elements, both schemes reach quickly the theoretical convergence order. In Figure 5 a direct comparison of the two methods is presented. The error is given as a function of the equivalent length scale (h), which depends on the number of total degrees of freedom ($nDOFs$): $h = (nDOFs)^{-0.5}$. The plot shows that the error for the ESR method is greater than the error for the BR2 method but, when the order is increased, the two schemes give comparable errors.

These numerical experiments were performed with a time marching approach in which explicit integration schemes were employed. The stability limits for the ESR method and the BR2 method are reported in Table VI. It is interesting to see that the allowed time step for the ESR method is approximately 4 times larger than the allowed time step for the BR2 method when $p = 1$. The ratio tends to grow when the order is increased and it is near 8 for $p = 6$. This behavior, together with the fact that for high orders the two schemes give comparable errors, suggest that the ESR method becomes competitive for explicit time integration of high order reconstructions. On the other hand, the fact that the BR2 method gives a lower error for all the reconstruction orders considered here, makes it a general purpose scheme, with a wider applicability than the ESR method.

It is very hard to compare the efficiencies of the different schemes. A possible approach might be to evaluate the total computational cost required to obtain a certain error level. However, the CPU time obtained is strongly related to the particular implementation chosen, and to the nature of the problem under study. Indeed, the presence of memory limitations (typically in 3D) or deforming grids can dramatically increase the computational cost, since several quantities cannot be stored but have to be re-computed during the simulation. Furthermore, the choice of the time integration scheme can enhance or cancel some benefits of the different methods. For example, in steady state problems, implicit integration schemes are commonly used. Their large stability domain can reduce the advantages observed in the explicit time integration with the ESR method.

Figure 5. L^2-norm of the temperature error vs the equivalent length scale h.

7.4. Considerations on implicit time integration

In the previous paragraph, explicit time integration schemes have been considered in order to evaluate the stability limits of the different methods. This aspect is particular important in time dependent problems. However, implicit schemes can be much more efficient than explicit ones when steady problems are studied. In this Section we describe how to implement the ESR method with implicit time integration. Implicit schemes require the evaluation of the jacobian of the system. If an approach based on the analytical computation of the jacobian is used, the ESR method shows some interesting features which can be useful during the implementation. A linear heat diffusion problem with a fixed mesh is considered as example.

In order to evaluate each term of the jacobian it is necessary to perform the derivative of the residuals with respect to each degree of freedom. This derivative contains contributions from the volume and the boundary integrals which appear in Eq. (14). In the proposed approach, the gradients required by the volume integral are approximated using the internal reconstruction. This simplifies significantly the computation of the jacobian. In contrast, both the RDG1x method and the BR2 method compute these gradients using information also from the interfaces. As far as the boundary integral is concerned, there are two possible cases. In fact, an interface can be inside the domain or it can belong to the domain boundary. In the first case, the required gradients are computed with the recovery approach. This is done through matrix-vector multiplications in which all the matrices depend only on the mesh and not on the solution:

$$\nabla u = [\mathbf{D}]\tilde{\mathbf{u}}^{\mathbf{r}} = [\mathbf{D}][\tilde{\mathbf{R}}] \left\{ \begin{array}{c} \tilde{\mathbf{u}}^{\mathbf{A}} \\ \tilde{\mathbf{u}}^{\mathbf{B}} \end{array} \right\}. \tag{39}$$

The terms in the jacobian related to this contribution can be immediately computed from the constant terms in the previous relation. On the other hand, if the interface belongs to a Dirichlet boundary, the procedure of Section 5 has to be considered. In particular the gradient on the boundary is computed as:

$$\nabla u = [\mathbf{D}] \left(\tilde{\mathbf{u}} + \Delta\tilde{\mathbf{u}} \right) = [\mathbf{D}]\tilde{\mathbf{u}} + [\mathbf{D}][\hat{\mathbf{B}}]\hat{\mathbf{W}}. \tag{40}$$

Again, the contribution to the jacobian can be directly obtained from the constant terms which appear in the previous relation.

A performance test was carried out on the 2D heat diffusion problem of Section 7.3. In particular an unstructured mesh with 512 triangles is considered. All the elements are fifth order accurate ($p = 4$). The solution is initialized with a uniform field and time integration is performed with both the RK4-LS explicit scheme and the backward Euler (BE) implicit scheme. In Figure 6 the L^2 norm of the residuals is reported as a function of the computational time. As expected, the plot shows the significant advantage offered by the implicit approach for steady problems.

Figure 6. Adimensionalized L^2-norm of the residual vs computational time [s].

8. P-ADAPTIVITY

The local nature of the discontinuous Galerkin reconstruction makes it easy to implement p-adaptive strategies. This is particularly true when a hierarchical basis is employed. In this case, when an element is refined, it is sufficient to add more degrees of freedom without the need to change the ones already known. On the other hand, if an element needs to decrease the reconstruction order, it is enough to eliminate the degrees of freedom related to the higher order modes. Several works are concerned with both h-adaptive and p-adaptive algorithms for discontinuous Galerkin methods. An overview of some techniques is presented in the book summarizing the results of the ADIGMA European project [36].

In this work, a very simple p-adaptive algorithm has been implemented in order to test the behavior

of the ESR method when the reconstruction order varies from an element to its neighbours. The algorithm is based on a sensor and a control law. The choice of the sensor depends on the particular set of equations which is considered. Its goal is to detect some features of the solution which can identify the most interesting regions. For example it can be related to the intensity of a variable or to its gradient. The control law is responsible of choosing the element order p_e according to the signal obtained locally by the sensor s_e. We use a simple control law with the following form:

$$p_e = \text{Int} \left[p_{min} + (p_{max} - p_{min}) \left(\frac{s_e - s_{min}}{s_{max} - s_{min}} \right)^{\alpha} \right] \tag{41}$$

where the function Int rounds its argument to the nearest integer number.

Eq. (41) shows that the reconstruction order in a generic element is allowed to vary between a minimum (p_{min}) and a maximum (p_{max}). The local value of the sensor (s_e) is compared to the minimum (s_{min}) and maximum (s_{max}) values founded in the computational domain at the considered time step. The coefficient α is a positive number which controls the distribution of the order in the domain: the lower the α the larger the number of elements in which high order reconstructions are employed. The chosen law helps in avoiding strong jumps in the order between neighbouring elements. The algorithm just described is not optimized, and we refer to the literature for more efficient approaches (for example [36]), but the main purpose of this Section is to evaluate the behavior of the ESR method, when the reconstruction order varies from an element to its neighbours. The implementation of better adaptive algorithms is left for future work.

8.1. Heat equation with p-adaptivity

We performed numerical experiments on the 2D heat equation with the p-adaptive algorithm. The influence of the parameter α was also investigated. The test case considered now is similar to the one used before for the convergence analysis, but now the number of semi-waves is increased up to 5 ($k = 5$) as shown in Figure 7. The exact solution is characterized by regions in which there are large gradients and regions in which the temperature distribution is close to constant. In a situation like this, the p-adaptive algorithm can show its advantages with respect to a uniform order discretization. The same unstructured meshes used in the previous Section were adopted for these experiments. We chose to use the magnitude of the temperature gradient as sensor variable.

Figure 7. Temperature field for the 2D heat equation problem, with $k = 5$.

Figure 8. Example of order distribution with p-adaptivity and the 8x8x2 mesh ($\alpha = 0.3$)

Figure 9. L^2-norm of the temperature error vs total number of DOF's (possible orders: $p_e = 1, 2, 3, 4, 5$).

Figure 10. L^2-norm of the temperature error vs total number of DOF's (possible orders: $p_e = 1, 3, 5$)

In the plot of Figure 9 the L^2-norm of the temperature error is reported as a function of the total number of degrees of freedom. DG1 and DG5 denote data obtained with a uniform order of accuracy, namely 1 and 5. The other data are obtained with the p-adaptive strategy (with $p_{min} = 1$ and $p_{max} = 5$). The control law defined by Eq.41 is used for this test. It can be seen that the p-adaptive algorithm is convenient with respect to the uniform order discretization if a sufficiently low value of α is chosen. The results suggest that the adaptive algorithm is competitive for $0.1 \le \alpha \le 0.2$.

We performed also a test in order to study the effects of the proposed discretization for the case $|p_A - p_B| > 1$. In particular, we modified the control law (Eq.41) in such a way that the algorithm can choose only the following reconstruction orders: $p_e = 1$, $p_e = 3$ or $p_e = 5$. In this way, when there is a jump in the order across an interface, the jump is always greater or equal than 2. The results for this somewhat artificial case are reported in Figure 10. A comparison with Figure 9 shows clearly that the presence of strong jumps in the order distribution is not convenient. For this reason, it is important to use a control law which chooses a smooth distribution of the order and which discourages the occurrence of the case $|p_A - p_B| > 1$.

9. EXPERIMENTS WITH THE NAVIER-STOKES EQUATIONS

In this Section the ESR method with the minimization BC's procedure (Section 5.2) is employed for the computation of viscous fluxes in the compressible Navier-Stokes equations. The convective fluxes are computed using the approximate Riemann solver due to Osher and Solomon [21] and Pandolfi [22]. In all tests an ideal gas with constant specific heat ratio $\gamma = 1.4$ and Prandtl number $Pr = 0.72$ is considered. All the meshes used in the simulations were produced with Gmsh [20] with the exception of the grid used for the tests of Section 9.1.

Three test cases were studied in order to compare our results with those available in the literature and with experimental data.

9.1. Laminar boundary layer on a flat plate

This test case was proposed at the first and second edition of the International Workshop on High-Order CFD Methods ([37], [38] and [39]). The flow is characterized by a Reynolds number based on the plate length equal to $Re_\infty = 10^6$. The angle of attack is $0°$ and the free-stream Mach number is $M_\infty = 0.5$. The plate is adiabatic and has unit length.

The purpose of this test case is to evaluate the behavior of the ESR method in the presence of highly clustered meshes. In particular we used one of the meshes proposed by Bassi and Colombo which can be downloaded from the Workshop website (file *a1-125-2s.msh*, [39]). This grid contains 560 quadrilateral elements. The distance between the inlet and the leading edge is 1.25. The distance between the freestream and the plate is 2. The size of the elements near the wall is $3.75 \cdot 10^{-4}$ in the normal direction. The most stretched element in the domain has a length/thickness ratio equal to 877.

We performed a p-refinement study on this mesh in order to evaluate the convergence of the solution. The simulations make it possible to compare the results with those presented by several groups at the workshop, which are reported in [38]. In Figure 11 the wall friction coefficient distribution is shown for several orders of accuracy. Also, Blasius' incompressible solution is reported. The effect of p-refinement can be easily seen in the region near the leading edge. We performed a second set of simulations with a lower free-stream Mach number ($M_\infty = 0.2$) in order to understand the influence of compressibility. In Table VII the drag coefficient (C_D) is reported as a function of the total number of degrees of freedom per equation ($nDOFs/eq$). The results show that, in this problem, compressibility does not have a significant influence. Even when the computations were performed using one of the meshes with less elements between those proposed at the Workshop, the p-refinement quickly obtained a converged solution. The ESR method behaves well also in the presence of highly clustered meshes with very stretched elements.

Figure 11. Wall friction coefficient (c_f) for the flat plate ($Re_\infty = 10^6, M_\infty = 0.5$)

9.2. Unsteady viscous flow over tandem airfoils

This 2D test case was proposed at the First and Second edition of the International Workshop on High-Order CFD Methods ([37], [38] and [39]). It is characterized by the presence of two

NACA0012 airfoils in tandem configuration. The leading airfoil is rotated by $10°$ about its aerodynamic center (25% of the chord) while the trailing airfoil is translated backward by 1.5 chords. The far field boundary is a circle with a radius of 100 chords centered on the leading edge of the trailing airfoil. The free-stream Mach number is $M_\infty = 0.2$ and the Reynolds number based on the chord of an airfoil is $Re_\infty = 10^4$. An initial smooth condition is prescribed according to the following relations:

$$v(x,y) = 0 \qquad P(x,y) = P_\infty \qquad \rho(x,y) = \rho_\infty$$
$$u(x,y) = \begin{cases} u_\infty & d > \delta \\ u_\infty \sin(\frac{\pi d}{2\delta}) & d \le \delta \end{cases}, \tag{42}$$

where u and v are the cartesian velocity components, d is the distance to the nearest wall and δ is a constant equal to 5% of the chord length.

We used two mixed structured-unstructured grids with 2101 and 8404 elements. Quadrilaterals are used near the airfoils and triangles in the rest of the domain. Elements with cubic edges are employed at solid walls. Here the size of the first layer of elements is equal to 0.005 chords in the direction normal to wall for the finest mesh. Figure 12 shows a detail of the finest mesh near the two bodies.

Figure 12. Detail of the finest mesh used for the tandem airfoils test.

The drag coefficient of the trailing airfoil was monitored during the first 20 convective times. We performed several simulations changing the reconstruction order from DG1 to DG5 on both meshes. Time integration is performed with RK4LS. The flow field is characterized by the stall of the leading airfoil. The vortices which are generated in the separation region strongly interact with the trailing airfoil.

A plot of the entropy field at the end of the simulation is reported in Figure 13 for DG5 on the finest mesh. The time evolution of the drag and lift coefficient for the trailing airfoil is reported in Figure 14 and Figure 15 for $p = 3, 4, 5$ on the finest mesh.

It is clear from these results that the flow is not periodic, at least in the time interval considered here. The chaotic aperiodic evolution of the vortices makes it difficult to perform a deterministic study. The results obtained by the different simulations tend to overlap for the first part of the computation but after a certain time they follow a different evolution. It can be seen that when the number of degrees of freedom is increased (with both mesh refinement or order refinement) the time interval in which the different solutions overlap becomes more extended. In particular the plots show that the DG4 method follows well the DG5 solution up to $time = 11$. In contrast, the DG3 prediction follows the higher order solutions only up to $time = 5$. A similar behavior can be seen in the results reported by other groups on the website of the First edition of the Workshop [38]. This test case showed that the proposed discretization is able to deal with strong unsteady wall-vortex interactions.

Figure 13. Entropy field for the tandem airfoils test at the end of the simulation (DG5RK4LS finest mesh)

Figure 14. Drag coefficient for the trailing airfoil (finest mesh p=3,4,5).

Figure 15. Lift coefficient for the trailing airfoil (finest mesh p=3,4,5).

9.3. Vortex shedding around a circular cylinder

In this classical test case, the unsteady flow field around a circular cylinder with $Re_\infty = 10^2$ and $M_\infty = 0.2$ is considered. Characteristic far field boundary conditions are prescribed on a circular

boundary. The distance of the far field from the body is equal to 100 cylinder diameters. The solid wall is adiabatic. The domain is discretized with a structured O-mesh with 19x48 quadrilateral elements. Wall elements have quartic edges.

Five simulations were performed with schemes ranging from DG1 to DG5 in order to evaluate the behavior of the proposed discretization in the presence of unsteady flow fields. Computations were carried out until periodicity was reached. The Strouhal number (St) related to vortex shedding and the average drag coefficient were evaluated over a period of 10 cycles. After that, the p-adaptive algorithm was tested and the order of the elements was allowed to vary between $p_{min} = 1$ and $p_{max} = 5$ at each time step.

First of all, some experiments with the algorithm's sensor were performed. Figure 16 and Figure 17 show the order distribution in the region near the cylinder when the magnitude of the velocity gradient or the entropy are used as sensors. In the first case, it can be seen that the separated flow zone does not have strong velocity gradients, and as a consequence the algorithm chooses to use low order reconstructions here. On the other hand, if entropy is used as sensor, the wake and the separation zone are correctly detected, but low order reconstructions are selected in front of the cylinder. For this reasons we chose to adopt a mixed approach: both sensors were used and the order of the element was set to the maximum between the orders suggested by the two sensors (see Figure 18). In all cases the control law described by Eq. (41) with $\alpha = 0.3$ was employed.

Table VIII reports the results obtained as a function of the total number of degrees of freedom per equation. Note that in the case of p-adaptivity the reported number of degrees of freedom refers to the range spanned in a period. The average drag coefficient and the Strouhal number are in line with the experimental data available in the literature (see Table IX). It can be seen from these results that the strong clustering of the mesh makes it difficult to resolve the wake in regions far from the cylinder. The natural remedy for this problem is the use of a hybrid hp-adaptive algorithm. Future work will be devoted to this aspect.

Figure 16. Distribution of the reconstruction order with the entropy sensor.

Figure 17. Distribution of the reconstruction order with the sensor based on velocity gradients.

Figure 18. Distribution of the reconstruction order using both the entropy and the velocity gradient sensors.

Figure 19. Entropy field around a circular cylinder with p-adaptivity ($Re_\infty = 10^2$).

10. CONCLUSIONS

In this paper, the ESR (Enhanced Stability Recovery) method for the computation of diffusive fluxes in the framework of discontinuous Galerkin methods is introduced and tested. The scheme follows the recovery approach [12] but it is based on a different recovery basis, with respect to the recovery basis required in the original recovery scheme (RDG1x). The RDG1x and the ESR methods seem to be complementary: the RDG1x method gives excellent results in pure diffusion problems while the ESR method finds its natural application in convection-diffusion problems on irregular meshes. The ESR method can be easily adopted with both structured and unstructured meshes thanks to the compactness of the reconstruction. The particular recovery basis proposed for the ESR method is flexible and makes the scheme robust since it remains well conditioned even in the presence of high order reconstructions on irregular meshes.

The accuracy of the proposed discretization has been compared with the accuracy of the widely used BR2 method [2]. Results show that the BR2 method is more accurate than the ESR method for low order reconstructions, but the two schemes become comparable for high order reconstructions. A convergence rate of $p + 1$ is obtained for the ESR method on unstructured meshes. The performed numerical experiments indicate that the stability limit with explicit time integration is significantly larger for the ESR method than the stability limits obtained with both the RDG1x and the BR2 schemes. In particular a comparison with the BR2 method on a 2D diffusion problem shows that this benefit grows when the order of the reconstruction is increased.

These features make the ESR method interesting for those applications in which explicit time integration is used in the presence of high order reconstructions. In particular, the method appears to be well suited for problems characterized by rigid bodies in which the matrices related to the recovery procedure can be computed and stored at the beginning of the simulation. On the other hand the method can become inefficient in the presence of deforming domains, due to the cost associated to the update of the recovery matrices during the simulation. In this kind of problems other methods which do not need the recovery procedure (for example BR2) can be more convenient.

As fas as the implementation is concerned, it is interesting to observe that the volume integral of the DG formulation is computed in a very simple way. Indeed, in the ESR method the gradient required for the volume integral is directly obtained by the internal element solution without the need to use information from neighbouring elements. If an implicit time integration scheme is used, this feature helps to simplify the analytical evaluation of the jacobian as it is described in Section 7.4.

The recovery procedure requires to introduce two different sets of basis functions, one for the element solution and one for the recovery solution. However, if the particular basis proposed in this work are employed, the same orthonormalization subroutines can be used for the construction of both the element and the recovery basis.

Further, a procedure for the weak imposition of Dirichlet boundary conditions is proposed. This procedure is based on a minimization algorithm and it can be used in the framework of recovery-based methods when curvilinear boundary elements are present. It does not need ghost elements which can be problematic in the presence of curvilinear boundaries. The algorithm is truly local thanks to the fact that it works on the element internal reconstruction and so it does not require data from neighbouring cells. In contrast, the original boundary condition procedure for the RDG1x method works on the recovery reconstruction and therefore it needs information also from the neighbouring elements.

The ESR method and the weak boundary conditions procedure were tested on 2D Navier-Stokes problems with flow separation. They perform well even in the presence of highly clustered meshes with very stretched elements such as those used in high Reynolds number flows. A simple p-adaptive algorithm was also successfully tested to show the ability of the ESR method to deal with elements of different order inside the domain. The numerical results were compared with both theoretical and experimental data and good agreement was found. Several tests considered in this work are by now benchmark tests, designed to assess the capabilities of numerical schemes for Navier-Stokes computations ([37],[38],[39]). Thus the numerical solutions obtained here can be easily compared with results available in the latest literature.

The behavior of the scheme in the presence of non-conforming meshes and the extension to 3D will be investigated in future work.

REFERENCES

1. Lo M, van Leer B. Analysis and Implementation of Recovery-Based Discontinuous Galerkin for Diffusion, *AIAA Paper 2009-3786* 2009.
2. Bassi F, Rebay S, Mariotti G, Pedinotti S, Savini M. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In: Decuypere R, Dibelius G, editors. *2nd European Conference on*

Turbomachinery Fluid Dynamics and Thermodynamics, Technologisch Institut, Antwerpen, Belgium, 1997, 99-108.

3. Arnold DN, Brezzi F, Cockburn B, Marini D. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis* 2002; **39**, 5 : 1749-1779.

4. Lions JL. Problemes aux limites non homogenes a donees irregulieres: une methode d'approximation, *Numerical Analysis of Partial Differential Equations (C.I.M.E. 2 Ciclo, Ispra, 1967)*. Edizioni Cremonese: Rome, 1968; 283-292.

5. Nitsche JA. Uber ein Variationsprinzip zur Losung Dirichlet-Problemen bei Verwendung von Teilraumen, die keinen Randbedingungen unteworfen sind. *Abh. Math. Sem. Univ. Hamburg* 1971 **36** : 9-5.

6. Babuska I, Zlamal M. Nonconforming elements in the finite element method with penalty, *SIAM Journal on Numerical Analysis* 1973; **10** : 863875.

7. Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations, *Journal of Computational Physics* 1997; **131** : 267-279.

8. Cockburn B, Shu CW. The local discontinuous Galerkin method for time-dependent convection-diffusion systems, *SIAM Journal on Numerical Analysis* 1998; **35** : 2440-2463.

9. Baumann CE, Oden JT. A discontinuous hp finite element method for convection-diffusion problems. *Comp. Meth. Appl. Mech. Eng.* 1999; **175** : 311-341.

10. Peraire J, Persson PO. The compact discontinuous Galerkin (CDG) method for elliptic problems, *SIAM Journal on Scientific Computing* 1999; **30** : 1806-1824.

11. Gassner G, Lorcher F, Munz CD., A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes, *Journal of Computational Physics* 2007; **224** : 1049-1063.

12. van Leer B, Nomura S. Discontinuous Galerkin for Diffusion, *AIAA paper 2005-5108* 2005.

13. Dumbser M, Zanotti O. Very High Order PNPM Schemes on Unstructured Meshes for the Resistive Relativistic MHD Equations, *Journal of Computational Physics* 2009; **228** : 6991-7006.

14. Luo H, Luo L, Nourgaliev R, Mousseau VA. A Reconstructed Discontinuous Galerkin Method for the Compressible Navier-Stokes Equations on Arbitrary Grids, *Journal of Computational Physics* 2010; **229**: 69616978.

15. Dumbser M, Balsara DS, Toro EF, Munz CD. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. *Journal of Computational Physics* 2008; **227** : 8209-8253.

16. Huynh HT. A reconstruction approach to high-order schemes including Discontinuous Galerkin for diffusion, *AIAA Paper 2009-0403* 2009.

17. Borrel M, Ryan J. The Elastoplast Discontinuous Galerkin (EDG) method for the Navier-Stokes equations, *Journal of Computational Physics* 2012, **231**: 1-22.

18. French DA, Galbraithy MC, Osorio M. Error Analysis of a Modified Discontinuous Galerkin Recovery Scheme for Diffusion Problems, *AIAA Paper 2010-1071* 2010.

19. van Leer B, Nomura S, van Raalte M. A discontinuous Galerkin Method for Diffusion Based on Recovery, *AIAA Paper 2007-4083* 2007.

20. Geuzaine C, Remacle JF. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* 2009; **79**: 1309-1331.

21. Osher S, Solomon F. Upwind schemes for hyperbolic systems of conservation laws, *Mathematics of Computation* 1982, **38**:339.

22. Pandolfi M. A contribution to the numerical prediction of unsteady flows, *AIAA Journal* 1984, **22**: 602-610.

23. Bassi F, Botti L, Colombo A, Di Pietro DA, Tesini P. On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations, *Journal of Computational Physics* 2012; **231** : 45-65.

24. Solin P, Segeth K, Dolezel I. *Higher-Order Finite Element Methods*, Chapman and Hall/CRC Press, 2003.

25. Oñate E. *Structural Analysis with the Finite Element Method: Linear Statics, Volume 1: Basis and Solids*. Springer: 2009;158-186.

26. Gottlieb S, Shu CW. Total variation diminishing Runge-Kutta schemes, *Mathematics of Computation* 1998; **67** : 73-85.

27. Carpenter MH, Kennedy C. Fourth-order 2N-storage Runge-Kutta schemes, *NASA Report TM 109112*, NASA Langley Research Center, 1994.

28. van Leer B, Lo M. Unification of Discontinuous Galerkin Methods for Advection and Diffusion, *AIAA Paper 2009-400* 2009.

29. Nourgaliev R, Theofanous T, Park H, Mousseau V, Knoll D. Direct Numerical simulation of interfacial flows, *AIAA Paper 2008-1453* 2008.

30. van Leer B, Lo M, Gitik R, Nomura S. *Adaptive High-order Methods in Computational Fluid Dynamics*. World Scientific: Singapore; 185-201.

31. Eaton JW, Bateman D, Hauberg S. *GNU Octave version 3.0.1 manual: a high-level interactive language for numerical computations*. CreateSpace Independent Publishing Platform: 2009.

32. Riviére B. *Discontinuous Galerkin methods for solving elliptic and parabolic equations : theory and implementation*. SIAM: Philadelphia, 2008; 66.

33. Bazilevs Y, Hughes TJR. Weak imposition of Dirichlet boundary conditions in fluid mechanics, *Computers & Fluids* 2007; **36** : 12-26.

34. Sudirham JJ, van der Vegt JJW, van Damme RMJ. A study on discontinuous Galerkin finite element methods for elliptic problems, *Memorandum No. 1690*, Department of Applied Mathematics, University of Twente

35. Bassi F, Rebay S. Numerical Evaluation of Two Discontinuous Galerkin Methods for the Compressible Navier-Stokes Equations. *International Journal for Numerical Methods in Fluids* 2002, **40** : 197-207.

36. Kroll N, Bieler H, Deconinck H, Couaillier V, van der Ven H, Sorensen K. ADIGMA - a European initiative on the development of adaptive higher-order variational methods for aerospace applications. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, vol. 113. Springer, 2010.

37. Wang ZJ, Fidkowski K, Abgrall R, Bassi F, Caraeni D, Cary A, Deconinck H, Hartmann R, Hillewaert K, Huynh HT, Kroll N, May G, Persson PO, van Leer B, Visbal M. High-order CFD methods: current status and perspective *International Journal for Numerical Methods in Fluids* 2013; **72**: 811-845.

38. First International Workshop on High-Order CFD Methods. http://zjwang.com/hiocfd.html (February 2014)
39. Second International Workshop on High-Order CFD Methods. http://www.dlr.de/as/hiocfd (February 2014)
40. Norberg C. Fluctuating lift on a circular cylinder: review and new measurements, *Journal of Fluids and Structures* 2003; **17** : 57-96.
41. Tritton DJ. Experiments on the flow past a circular cylinder at low reynolds number, *Journal of Fluid Mechanics* 1959; **6** : 547-567.
42. von Wieselsberger C. Neuere Festellungen ber die Gesetze des Flssigkeits un Luftwiderstands, *Phys. Z.* 1921; **22** : 321-328.
43. Williamson CHK. Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low reynolds number, *Journal of Fluid Mechanics* 1989; **206** : 579-627.

Scheme	Cond. number without ON	Cond. number with ON
DG1	4.61E+02	1.97E+00
DG2	1.05E+04	3.87E+00
DG3	2.52E+05	5.36E+00
DG4	6.25E+06	6.62E+00
DG5	1.58E+08	7.87E+00
DG6	5.03E+09	9.09E+00

Table I. Condition number of the matrix $\left([\mathbf{R}]^T[\mathbf{R}]\right)$ for the interface of Figure 4 without and with orthonormalization (ON) of the recovery basis

Scheme	$\nu\Delta t/\Delta x^2$
RDG1x (from [30])	0.08
BR2 (from [30])	0.10
ESR	0.165

Table II. 1D Poisson problem: stability limit (p=1, forward Euler)

Mesh	RDG1x L^2 error	Order	BR2 L^2 error	Order	ESR L^2 error	Order
8	8.90E-04	-	3.06E-02	-	5.48E-02	-
16	4.24E-05	4.39	8.67E-03	1.82	1.44E-02	1.93
32	2.18E-06	4.28	2.25E-03	1.95	3.69E-03	1.96
64	1.21E-07	4.17	5.66E-04	1.99	9.35E-04	1.98

Table III. 1D Poisson problem: L^2 errors and convergence order (p=1)

Mesh	RDG1x L^2 error	BR2 L^2 error	ESR L^2 error
5	5.58E-03	1.39E-02	1.77E-02
10	1.58E-03	4.25E-03	6.08E-03
20	4.25E-04	1.56E-03	1.82E-03

Table IV. 1D Convection-diffusion problem: L^2 errors at steady state (p=1)

Scheme	Mesh	Error ESR	Order ESR	Error BR2	Order BR2
DG1	$2 \times (2 \times 2)$	7.33E-02	-	2.98E-02	-
	$2 \times (4 \times 4)$	2.02E-02	**1.86**	1.23E-02	**1.28**
	$2 \times (8 \times 8)$	4.92E-03	**2.04**	3.88E-03	**1.66**
DG2	$2 \times (2 \times 2)$	2.15E-02	-	8.79E-03	-
	$2 \times (4 \times 4)$	2.00E-03	**3.43**	9.77E-04	**3.17**
	$2 \times (8 \times 8)$	2.47E-04	**3.02**	1.14E-04	**3.10**
DG3	$2 \times (2 \times 2)$	1.75E-03	-	1.08E-03	-
	$2 \times (4 \times 4)$	9.17E-05	**4.25**	7.38E-05	**3.87**
	$2 \times (8 \times 8)$	5.32E-06	**4.11**	4.79E-06	**3.95**
DG4	$2 \times (2 \times 2)$	1.45E-04	-	1.23E-04	-
	$2 \times (4 \times 4)$	4.90E-06	**4.89**	4.25E-06	**4.86**
	$2 \times (8 \times 8)$	1.45E-07	**5.08**	1.41E-07	**4.91**
DG5	$2 \times (2 \times 2)$	1.51E-05	-	1.24E-05	-
	$2 \times (4 \times 4)$	2.96E-07	**5.67**	2.12E-07	**5.87**
	$2 \times (8 \times 8)$	4.53E-09	**6.03**	3.44E-09	**5.95**
DG6	$2 \times (2 \times 2)$	1.25E-06	-	8.82E-07	-
	$2 \times (4 \times 4)$	9.74E-09	**7.00**	7.60E-09	**6.86**
	$2 \times (8 \times 8)$	7.79E-11	**6.97**	6.15E-11	**6.95**

Table V. L^2-norm of the temperature error for the ESR and the BR2 methods.

Scheme	σ_{max}ESR	σ_{max} BR2
DG1-RK2TVD	6.2E-02	1.4E-02
DG2-RK3TVD	3.9E-02	8.5E-03
DG3-RK4LS	5.6E-02	8.6E-03
DG4-RK4LS	3.5E-02	5.0E-03
DG5-RK4LS	2.4E-02	3.2E-03
DG6-RK4LS	1.6E-02	2.1E-03

Table VI. Stability limit with explicit time integration for 2D heat diffusion on unstructured meshes.

Scheme	nDOFs/eq	C_D ($M_\infty = 0.5$)	C_D ($M_\infty = 0.2$)
DG1-RK2TVD	1680	1.283	1.282
DG2-RK3TVD	3360	1.324	1.324
DG3-RK4LS	5600	1.327	1.327
DG4-RK4LS	8400	1.327	1.327
DG5-RK4LS	11760	1.327	1.327
Blasius	-	1.328	1.328

Table VII. Drag coefficient (C_D) for flate plate problem ($Re_\infty = 10^6, M_\infty = 0.5 - 0.2$).

Scheme	nDOFs/eq	Average C_D	St
DG1-RK2TVD	2736	1.287	0.137
DG2-RK3TVD	5472	1.334	0.161
DG3-RK4LS	9120	1.334	0.163
DG4-RK4LS	13680	1.336	0.163
p-adapt.-RK4LS	5200-5300	1.335	0.163

Table VIII. Average drag coefficient and Strouhal number for cylinder flow ($Re_\infty = 10^2, M_\infty = 0.2$).

Author	Average C_D	St
Norberg [40]	-	0.164
Tritton [41]	1.26	0.157-0.164
Wieselsberger [42]	1.43	-
Williamson [43]	1.33	0.160-0.164

Table IX. Experimental results from the literature for cylinder flow ($Re_\infty = 10^2$)